

Computationally efficient Brownian dynamics via wavelet Monte Carlo

Oliver T. Dyer¹ and Robin C. Ball¹

Department of Physics, University of Warwick, Coventry, CV4 7AL, UK

(Dated: 29 November 2016)

We develop a Brownian dynamics algorithm which evolves soft matter systems by spatially correlated Monte Carlo moves rather than the usual decomposition of the mobility tensor. The algorithm uses vector wavelets as its basic moves and produces hydrodynamics in the low Reynolds number regime according to the Oseen tensor. When small moves are removed the correlations instead closely approximate the Rotne-Prager tensor, itself widely used to correct for deficiencies in Oseen. We also include plane wave moves to provide the longest range correlations, which we detail for systems in both an infinite and periodic box. The computational cost of the algorithm scales competitively with system size and has a small prefactor due to the method's simplicity. Homogeneous systems of N particles and fixed concentration exhibit cost scaling of $N \ln N$ that, in a rough comparison, would only become more expensive than an established lattice Boltzmann algorithm when N is many orders of magnitude larger than in any currently feasible simulation. For dilute systems, the cost scales as N and the comparisons are even more favourable. We also validate the algorithm by checking it reproduces the correct dynamics in simple single polymer systems, as well as verifying the effect of periodicity on the mobility tensor.

I. INTRODUCTION

Brownian dynamics (BD) algorithms aim to simplify soft matter simulations by replacing the large number of degrees of freedom in the solvent with known hydrodynamic interactions (HIs), which simply need to be calculated between the N particles of interest at each time step.¹

This compares to explicit solvent algorithms that follow solvent molecules with some level of coarse graining, despite not being interested in them directly, to let them mediate viscosity and HIs through local interactions. Such methods include molecular dynamics (MD),² dissipative particle dynamics (DPD),³ lattice Boltzmann (LB)^{4–6} and multi-particle collision dynamics (MPCD)⁷ algorithms. The computational cost, or time taken to run a simulation evolving a system by some physical amount of time, of these methods scales linearly with the total number of particles. This includes particles in the solvent as well as those of interest. For systems of fixed concentration this leads to the cost scaling as N ,⁸ though the overhead cost of moving the solvent molecules limits the feasible system size, especially as the systems become more dilute. When considering a non-periodic system the scaling rises. The important example of single polymer chains leads to $N^{3\nu}$, with ν the Flory exponent, in order to fit the whole chain inside the simulation box.⁶

Despite reducing the number of degrees of freedom significantly, the cost of conventional BD algorithms, which is dominated by the decomposition of the mobility tensor, limits their ability to simulate large N systems. Fixman's algorithm is well known to cut the scaling of this decomposition down to $N^{2.25}$ from the naive N^3 ,⁹ and several methods have since been put forward to reduce the scaling further.^{8,10} It has even been reduced to or near $N \ln N$ in some cases, although these approaches are only valid for bounded systems,¹¹ or introduce errors to allow more efficient computation via particle mesh Ewald

techniques^{12,13} or sparse arrays.¹⁴

In this work we present an entirely different approach to BD, using a Monte Carlo (MC) algorithm to bypass explicit calculations with the mobility tensor altogether. To date, MC methods in the field have been used primarily to study equilibrium properties since they have not accounted for HIs. A variety of different particle movement schemes have been used, including simple methods moving individual particles¹⁵ and evolving the system with torsional rotations of bonds in polymers.^{16–18} So called bridging moves have also been introduced to handle branching polymers,^{19,20} and more recently 'event-chain' algorithms have been introduced to handle hard sphere particles.^{21,22}

An important bridge between the MC methods above and our hydrodynamically coupled method below lies in the work of Maggs.²³ Maggs introduced spatially extended correlated MC moves which he tuned to maximise the equilibration rate of a simple fluid system. He did not target HIs per se, but he was led to motion of the same scaling with arbitrary moves. We focus on moves described by wavelets, a class of function that has seen use in many areas of physics, particularly for signal processing because they form an (over) complete, localised basis that allows temporal changes in a signal to be identified.^{24,25} The development of wavelet theory is catalogued in Ref. 26. For this work, their role as basis functions enables us to re-express the mobility tensor in a form readily transferable to a MC simulation. We will show that this approach leads to a cost scaling that is at worst $N \ln N$ with a very competitive prefactor and no assumptions on the system beyond those already in basic BD algorithms.

In section II we begin with sketch calculations to highlight how HIs and the cost scaling arise without needing the full details. Section III addresses the background physics and mathematics required for the method before section IV explains the wavelet method in full. Section V

describes a necessary modification to include occasional plane wave moves, before results of simple validation simulations are given in section VI.

A. Units and potentials used in data

For the data given in this paper, but not required by the theory, we have used polymeric systems in a good solvent. The polymers are represented by bead-spring chains with finitely extensible non-linear elastic (FENE) springs and the Weeks-Chandler-Anderson (WCA) potential acting between all particles at positions \mathbf{r}_i :

$$U_{\text{FENE}} = -\frac{1}{2}k_{\text{FENE}}R_0^2 \ln(1 - (r_{ij}/R_0)^2), \quad (1)$$

$$U_{\text{WCA}} = 4\epsilon \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 + \frac{1}{4} \right), \quad (2)$$

for $r_{ij} = |\mathbf{r}_{ij}| = |\mathbf{r}_i - \mathbf{r}_j| < R_0, 2^{1/6}\sigma$ respectively.

We adopt as length and energy units $2^{1/6}\sigma$ and ϵ , and denote non-dimensionalised quantities with a bar. Therefore, to match physical systems in Ref.s 6 and 8 for benchmarking performance, we use $\bar{\sigma} = 2^{-1/6}$, $\bar{\epsilon} = 1$, $\bar{k}_{\text{FENE}} = 7 \times 2^{2/6}$ and $\bar{R}_0 = 2 \times 2^{-1/6}$.

Matching hydrodynamic radii of individual particles, a , leads to the minimum wavelet radius (introduced in section IV D) $\bar{\lambda}_{\text{min}} = 0.700$, while the thermal energy is $\bar{k}_BT = 1.2$.

We define our unit of time to be the time over which completely isolated or non-interacting particles are expected to have diffused by their own radius:

$$\tau = \pi\eta a^3/k_BT. \quad (3)$$

By requiring the same viscosity η as used in Ref.s 6 and 8, which will only appear implicitly in our algorithm, this unit of time is 4.04 times larger than in those papers.

Finally, to match the systems in Ref. 8, whenever a system is called ‘semi-dilute’ it consists of polymer chains of length $N_b = 10$ beads and a global bead concentration $N/\bar{L}^3 = 0.625$, where L is the side length of our simulation box.

II. METHOD SKETCH AND MOTIVATION

We seek a MC algorithm that produces hydrodynamics in a simple and efficient way. The moves considered are displacements inside spheres of radius λ centred on position \mathbf{b} , chosen from probability density functions (PDFs) \mathcal{P}_λ and $\mathcal{P}_{\mathbf{b}|\lambda}$ respectively. Here we have anticipated some λ -dependence in the distribution for \mathbf{b} . The orientation, $\hat{\mathbf{p}}$, of these moves is unbiased so that average motion is only induced by non-zero forces weighting the Metropolis test. An example of such a move is depicted in Fig. 1.

A move will only contribute to correlated motion of particles i and j if it encloses both of them, requiring

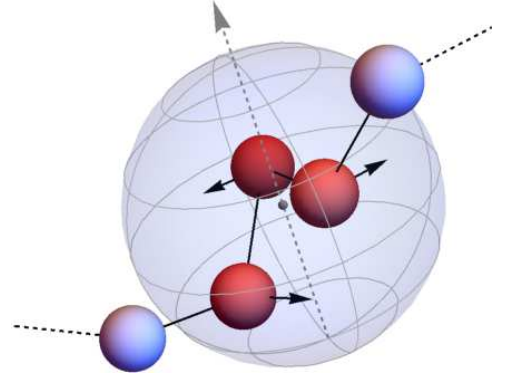


FIG. 1. Schematic diagram of a wavelet move on a section of polymer. The dashed arrow represents $\hat{\mathbf{p}}$ and the central dot marks \mathbf{b} . The particles enclosed in the wavelet move according to the vector $A_w\mathbf{w}$, introduced in section III A, while particles lying outside remain stationary.

$2\lambda \geq r_{ij}$ and for \mathbf{b} to land in a volume of order λ^d in d dimensions. The contribution from a given move therefore has the piecewise form

$$\langle \delta r_i \delta r_j \rangle \sim \begin{cases} 0 & \text{for } 2\lambda < r_{ij} \\ \mathcal{P}_{\mathbf{b}|\lambda} \lambda^d A_w^2 & \text{for } 2\lambda \geq r_{ij} \end{cases}, \quad (4)$$

where A_w is a displacement amplitude that is related to strain by $\epsilon \propto A_w/\lambda$ if, for simplicity in this section, the strain is assumed to be uniform over the move. In that case the estimate for the change in energy over a move containing n particles is $\Delta U \sim n\epsilon^2$, which we are constrained by a Metropolis test to avoid being large compared to k_BT . This sets $A_w \sim \lambda/\sqrt{n}$.

For $\mathcal{P}_{\mathbf{b}|\lambda}$ we note that moves with $n = 0$ contribute no particle motion and are desirably avoided altogether. Those with $n \geq 1$ can be chosen by first picking a particle at random, and then choosing \mathbf{b} at random within distance λ . $\mathcal{P}_{\mathbf{b}|\lambda}$ is then exactly proportional to $n/(N\lambda^d)$ since each of the n particles could have led to that centre being chosen.

Finally we pick $\mathcal{P}_\lambda \sim \lambda^{-d-1}$, so that integrating Eq. (4) over radii we have

$$\langle \delta r_i \delta r_j \rangle \sim \int_{r_{ij}/2}^{\infty} d\lambda \lambda^{-d+1} / N \propto r_{ij}^{2-d} / N, \quad (5)$$

matching the required form for hydrodynamics in d dimensions.

We now know the algorithm produces the desired dynamics so we turn our attention to estimating the computational cost to evolve our system by a physical amount of time. This will be equal to the cost per move divided by the time evolved per move. The time factor comes from the previous result, using Brownian motion to set it proportional to $r_{ij}^{2-d} \delta t$ so that we have a time evolution per move going as

$$\delta t \sim 1/N. \quad (6)$$

Meanwhile, the cost per move of radius λ is of order the expected number of particles involved $n(\lambda)$, so the increment in the computational cost is given by $\delta C = c \int d\lambda \mathcal{P}_\lambda n(\lambda)$, with c being the cost per particle. The homogeneous case has the cost integral diverging logarithmically, leading to

$$\frac{\delta C}{\delta t} \sim cN \ln N. \quad (7)$$

The factor of N here simply reflects the distribution of computational effort across the system, while the $\ln N$ will be seen in section IV F to stem from an increase in allowed λ values. It is also shown that for a fractal system, such as a single polymer chain, the logarithm is no longer present.

The method outlined above is therefore very simple, with a favourable cost scaling compared to conventional BD and without the solvent degrees of freedom of explicit solvent methods. The rest of this paper describes the method in detail for $d = 3$.

III. MOBILITY TENSORS

We focus on the low Reynolds number limit in which the dynamics of the fluid solvent is governed by the Stokes equations. For simplicity we assume we have spherical particles subject to no applied torques, so the dynamics in the system reduces to interrelating the particle velocities, \mathbf{v}_i , to the applied forces \mathbf{F}_j , the latter including interparticle forces. In general this is given by

$$\mathbf{v}_i = \sum_j \mathcal{G}_{ij} \cdot \mathbf{F}_j + \delta \mathbf{v}_i, \quad (8)$$

where \mathcal{G} is the mobility tensor and the superposability of the velocity response follows from the linearity of the Stokes equations. The contribution from random thermal fluctuations, $\delta \mathbf{v}_i$, have covariance set by the fluctuation dissipation theorem (FDT), which in the Stokes limit reduces to $\langle \delta \mathbf{v}_i(t) \otimes \delta \mathbf{v}_j(t') \rangle = 2k_B T \mathcal{G}_{ij} \delta(t - t')$. In practice we need to implement displacements across a small but non-zero time interval δt , in terms of which this becomes

$$\langle \delta \mathbf{r}_i \otimes \delta \mathbf{r}_j \rangle = 2k_B T \mathcal{G}_{ij} \delta t. \quad (9)$$

In general the mobility tensor depends on the entire configuration of particle positions and here we have to simplify, following other workers in using the leading dilute limit expressions. Thus for the self-terms $i = j$ we take the Stokes form $\mathcal{G}_{ii} = \mathbf{I}/6\pi\eta a$, where a is assumed monodisperse for simplicity of exposition. For the cross terms we have $\mathcal{G}_{ij}(\mathbf{r}_i, \mathbf{r}_j) = \mathbf{g}(\mathbf{r}_i - \mathbf{r}_j)$ when the interference of third particles is ignored, and then to leading order in powers of separation $r = |\mathbf{r}_i - \mathbf{r}_j|$ we obtain the Oseen tensor,²⁷

$$\mathbf{g}_{\text{Oseen}}(\mathbf{r}) = \mathcal{O}(\mathbf{r}) = \frac{1}{8\pi\eta r} (\mathbf{I} + \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}), \quad (10)$$

corresponding to particles comoving with the bare solvent response, and as a result $\nabla \cdot \mathbf{g}_{\text{Oseen}}(\mathbf{r}) = 0$.

As is well known, the Oseen tensor alone does not assure a mobility matrix with non-negative eigenvalues²⁸ (crucial for the FDT result) so some modification must always be taken at small r to remedy this. It has become standard in the recent literature to use the Rotne-Prager (RP) form.²⁹ This incorporates the next-leading power of distance, which also turns out to be divergence free:

$$\mathbf{g}_{\text{RP}}(\mathbf{r}) = \mathcal{O}(\mathbf{r}) + \frac{1}{12\pi\eta r} \left(\frac{a}{r}\right)^2 (\mathbf{I} - 3\hat{\mathbf{r}} \otimes \hat{\mathbf{r}}). \quad (11)$$

In our wavelet method we implement the Oseen tensor modified by a cut-off in the wavelet spectrum. This is positive definite and we show in section IV G can lead to a tensor very close to \mathbf{g}_{RP} . It should be noted that whilst these modifications render the total mobility tensor positive definite, they still underrepresent the reduction in mutual mobility of two particles at close approach, for which $\hat{\mathbf{r}} \cdot \mathbf{g} \cdot \hat{\mathbf{r}}$ should match the self term $1/6\pi\eta a$ as $r \rightarrow 2a$.

A. Wavelet representation of the Oseen tensor

Our method is an adaptation of the continuous wavelet transform in three dimensions, itself based on the identity for the Dirac delta function

$$\delta(\mathbf{r}_i - \mathbf{r}_j) = \mathcal{N}_\delta \int \frac{d\lambda}{\lambda} \frac{d^3\mathbf{b}}{\lambda^3} \frac{1}{\lambda^3} w\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}\right) w\left(\frac{\mathbf{r}_j - \mathbf{b}}{\lambda}\right). \quad (12)$$

The choice of ‘mother wavelet’ shape is surprisingly arbitrary, with real wavelets constrained only by the formal requirements²⁵

$$\int d^3\mathbf{x} w(\mathbf{x})^2 < \infty, \quad (13)$$

$$\int d^3\mathbf{x} w(\mathbf{x}) = 0. \quad (14)$$

The normalising front factor is then given by the wavelet Fourier transform $\tilde{w}(\mathbf{k})$: $\mathcal{N}_\delta^{-1} = (2\pi)^2 \int d^3\mathbf{k} k^{-3} |\tilde{w}(\mathbf{k})|^2$.

For this work it is advantageous to impose additional constraints. The first is to limit the support of w so that

$$w(\mathbf{x}) = 0 \quad \text{for } |\mathbf{x}| > 1. \quad (15)$$

Then it is clear that $w((\mathbf{r} - \mathbf{b})/\lambda)$ is non-zero only over a region of radius, or ‘scale’, λ centred on \mathbf{b} .

Next we modify Eq. (12) in two ways. First we make the wavelet a vector valued function, $\mathbf{w}((\mathbf{r} - \mathbf{b})/\lambda, \hat{\mathbf{p}})$, with additional input ‘polarisation’ variable, $\hat{\mathbf{p}}$. We require these wavelets to be divergence-free,

$$\nabla \cdot \mathbf{w}(\mathbf{x}, \hat{\mathbf{p}}) = 0, \quad (16)$$

so that \mathbf{g} inherits this property below. It is this constraint of the wavelet being transverse which forces us to supply $\hat{\mathbf{p}}$. Our vector wavelet can now be thought of as a flow field in the vicinity of \mathbf{b} extending over a lengthscale λ .

Our second modification is to change the explicit power of λ in Eq. (12) so that the dimensions match that of the Oseen tensor. This then leads us to consider

$$\mathbf{g}(\mathbf{r}_i - \mathbf{r}_j) = \mathcal{N}_g \int \frac{d\lambda}{\lambda} \frac{d^3\mathbf{b}}{\lambda^3} d^2\hat{\mathbf{p}} \times \frac{1}{\lambda} \mathbf{w}\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \otimes \mathbf{w}\left(\frac{\mathbf{r}_j - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right). \quad (17)$$

In Appendix A we show that this is exactly equal to $\mathcal{O}(\mathbf{r}_i - \mathbf{r}_j)$ when λ is integrated from 0 to ∞ and with an appropriate choice of the constant \mathcal{N}_g .

Eq. (17) is more enlightening when expressed as an expectation value. In doing so we introduce a wavelet amplitude A_w , which subsumes all constant factors, but may also depend on the wavelet parameters as well. This dependence is found in section IV C, but for now we assume it is known and write

$$\mathcal{O}(\mathbf{r}_i - \mathbf{r}_j) = \left\langle A_w \mathbf{w}\left(\frac{\mathbf{r}_i - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \otimes A_w \mathbf{w}\left(\frac{\mathbf{r}_j - \mathbf{b}}{\lambda}, \hat{\mathbf{p}}\right) \right\rangle_{\lambda, \mathbf{b}, \hat{\mathbf{p}}}, \quad (18)$$

where subscripts on the angle brackets indicate quantities averaged over. Finally, comparison between Eq.s (9) and (18) immediately identifies $A_w \mathbf{w}((\mathbf{r} - \mathbf{b})/\lambda, \hat{\mathbf{p}})$ as the displacement $\delta\mathbf{r}$.

IV. DESCRIPTION OF THE WAVELET METHOD

With the expectation value interpretation in Eq. (18), we do not need to compute the integral for each particle at every time step, and can instead sample wavelets with appropriate distributions for λ , \mathbf{b} and $\hat{\mathbf{p}}$. After a large number of wavelets have been chosen the evolution of the system will closely approximate what the Oseen tensor would have given.

The details of how this is implemented are given below, but the basic structure of the algorithm follows the process:

- 1: generate wavelet parameters from their distributions,
- 2: provisionally move particles according to the resulting wavelet,
- 3: calculate the energy change, ΔU , caused by this move and accept or reject it with a Metropolis probability

$$P_{\text{acc}} = \min(1, e^{-\Delta U/k_B T}). \quad (19)$$

The process is then repeated for a desired number of moves.

In the Metropolis test there is no need to include a term for the Jacobian of the move, as per Maggs,²³ because we will only consider moves with Jacobian equal to unity.

A. Choice of mother wavelet

The restrictions in Eq.s (13)-(16) still leave a choice for \mathbf{w} . In this article we satisfy these conditions by using the form

$$\mathbf{w}(\mathbf{r}, \hat{\mathbf{p}}) = \begin{cases} \hat{\mathbf{p}} \times \nabla \phi(\mathbf{r}) & \text{for } |\mathbf{r}| \leq 1 \\ \mathbf{0} & \text{for } |\mathbf{r}| > 1 \end{cases}, \quad (20)$$

for some scalar function ϕ . This provides significant simplification to results, especially from section IV G onwards.

We also choose to limit ourselves to continuous wavelets so that the strain tensor is finite everywhere. This is required by neither wavelet theory nor the algorithm, but does simplify some of the analysis. This leads us to consider the ‘tapered wavelet’ as our basic example, with which all explicit results given are obtained. In conventional spherical polar coordinates (r, θ, φ) , and polarisation vector along the z -axis ($\theta = 0$), this is given by

$$\mathbf{w}(\mathbf{r}, \hat{\mathbf{z}}) = r \sin \theta (1 - r) \hat{\varphi} \quad \text{for } r \leq 1. \quad (21)$$

The associated ϕ and its Fourier transform are

$$\phi(r) = \frac{1}{2}r^2 - \frac{1}{3}r^3 - \frac{1}{6} \quad \text{for } r \leq 1, \quad (22)$$

$$\tilde{\phi}(k) = 4\pi k^{-6} (5k \sin k - (k^2 - 8) \cos k - 8). \quad (23)$$

B. Distribution of wavelet centres

With the mother wavelet chosen, we now need to determine the distributions from which to pick the parameters. For $\hat{\mathbf{p}}$ we assume the system is isotropic so choose A_w to be independent of it, and its PDF is then simply $\mathcal{P}_{\hat{\mathbf{p}}} = 1/4\pi$.

The PDF for \mathbf{b} is more involved as we want to avoid spending CPU time on moves that contain no particles and hence don’t evolve the system of interest. To ensure all wavelets contain at least one particle we first pick a particle, all with equal probability, and then choose \mathbf{b} uniformly inside a sphere of radius λ centred on this particle.

This approach introduces biases that need to be accounted for. First, the probability of choosing a position inside a volume element is inversely proportional to the volume of the sphere. Similarly, the chance that any given particle is chosen is inversely proportional to N . Lastly, if the resulting wavelet contains n particles, there must have been n possible ways to have chosen it, and hence the probability is increased by this factor. All combined, we have the PDF for \mathbf{b}

$$\mathcal{P}_{\mathbf{b}|\lambda} = 3n/4\pi\lambda^3 N. \quad (24)$$

C. Choosing the wavelet amplitude

Before the PDF for λ can be determined, we need to know the form of A_w . To find this we note that if we want the distribution of λ to be correctly reflected in the accepted moves, we want P_{acc} , and hence ΔU , to be as constant as possible over all moves.

An estimate of ΔU can be made with the strain energy associated with a move

$$\Delta U = \frac{1}{2} \mu \frac{3n}{4\pi\lambda^3} \int_{\text{wavelet}} d^3\mathbf{r} (\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} + \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}^T), \quad (25)$$

where μ is the shear modulus, $\boldsymbol{\varepsilon}$ is the move's strain tensor, and $:$ denotes a double dot product. Assuming small displacements we can use the infinitesimal strain tensor $\boldsymbol{\varepsilon} = A_w (\nabla \mathbf{w} + (\nabla \mathbf{w})^T) / 2$.

With no preferred direction, n accounts for all location dependence so \mathbf{b} and $\hat{\mathbf{p}}$ can be left out of the calculation, leaving only the transformation $r \rightarrow r/\lambda$ from the mother wavelet. For the tapered wavelet, the only non-zero strain elements are $\varepsilon_{r\varphi} = \varepsilon_{\varphi r} = -A_w r \sin \theta / 2\lambda^2$, giving the energy estimate

$$\begin{aligned} \Delta U &= \frac{3}{16\pi} \mu n \lambda^{-7} A_w^2 \oint d\varphi \int_0^\pi d\theta \sin^3 \theta \int_0^\lambda dr r^4 \\ &= \mu n \lambda^{-2} A_w^2 / 10. \end{aligned} \quad (26)$$

For this to be constant over all wavelets we must choose

$$A_w = A_0 \lambda / \sqrt{n}, \quad (27)$$

where A_0 is a constant chosen to give energies of order $k_B T$.

Fig. 2 reveals that choosing A_w this way leads to constant P_{acc} only asymptotically, where it limits to a value P_{acc}^∞ . This is unavoidable since in the limit $\lambda \rightarrow 0$ wavelets don't actually change the system so must have $P_{\text{acc}} \rightarrow 1$. The existence of this asymptote guarantees ΔU is not routinely larger than $k_B T$ for any λ , but we still need to ensure the correct distribution is observed in the accepted moves. For this we use a λ recycling scheme in which the radius of a rejected move is reused until a move is accepted.

Also seen in Fig. 2 is the effect of particle density. The decrease in P_{acc} at higher densities comes from the increased chance of any pair of particles moving to a thermally forbidden separation. The asymptotic values can rise significantly when potentials are made softer or removed, for example when simulating a system in θ solvent. For fixed length polymers, the asymptote approaches a maximum in the dilute limit, where the density in the chain itself sets the effective density seen by wavelets.

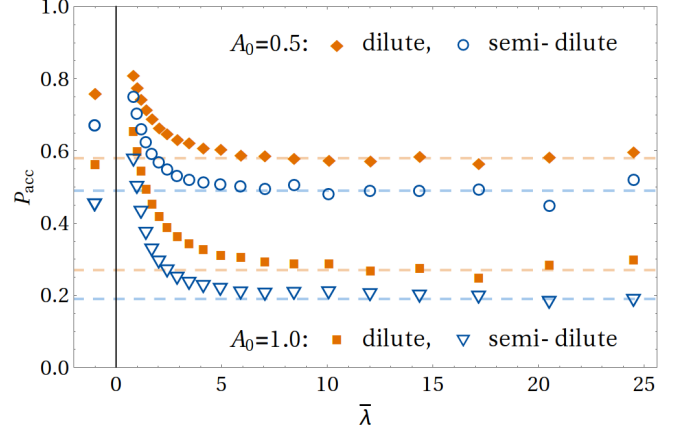


FIG. 2. Acceptance probabilities over the spectrum of wavelet radii at different move amplitudes. The dilute data used an isolated polymer chain with $N = 2048$. The dashed lines are only to emphasize the asymptotic behaviour, while the markers at $\bar{\lambda} = -1$ indicate the average P_{acc} over all wavelets.

D. Distribution of wavelet radii

With A_w^2 , $\mathcal{P}_{\hat{\mathbf{p}}}$ and $\mathcal{P}_{\mathbf{b}|\lambda}$ identified, we can now determine \mathcal{P}_λ from Eq. (17) by requiring

$$\lambda^{-5} \propto \mathcal{P}_\lambda \mathcal{P}_{\mathbf{b}|\lambda} \mathcal{P}_{\hat{\mathbf{p}}} A_w^2. \quad (28)$$

We therefore use the PDF

$$\mathcal{P}_\lambda = \mathcal{N}_\lambda \lambda^{-4}; \quad \mathcal{N}_\lambda = 3(\lambda_{\min}^{-3} - \lambda_{\max}^{-3})^{-1}, \quad (29)$$

when normalised between λ_{\min} and λ_{\max} . The effect of using these finite bounds instead of $(0, \infty)$ is discussed in section IV G. For now we simply comment that a finite λ_{\min} is desirable as it regularises the singularity in the Oseen tensor at $r \rightarrow 0$ and gives us a particle radius via $a = \lambda_{\min} / \lambda_a$, with λ_a a constant dependent only on the choice of mother wavelet. λ_{\max} is required when considering a simulation in a finite box of side length L , where we do not want wavelets to overlap with their periodic images due to the additional computational effort that introduces.

For a homogeneous system an alternative method can be used, whereby \mathbf{b} is chosen uniformly across the box so that $\mathcal{P}_{\mathbf{b}|\lambda} = 1/L^3$, independent of λ . In this case it is sufficient to consider the mean $\langle n \rangle$, using the global rather than local density. This leads to $A_w = A_0 L^3 / N \sqrt{\lambda}$ and, again, $\mathcal{P}_\lambda = \mathcal{N}_\lambda \lambda^{-4}$.

The rapid decay of \mathcal{P}_λ means small radius moves dominate. This is reflected in Fig. 2 with the global P_{acc} being significantly higher than the asymptote.

E. Time evolution per move

Next we need to know how much physical time passes during a simulation, for which we calculate the expected

displacement squared of any given particle in a single move, $\langle \delta r_i^2 \rangle_1$. This simplifies to the integral

$$\langle \delta r_i^2 \rangle_1 = \int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \mathcal{P}_\lambda \int_{\text{wavelet}} d^3\mathbf{r} \mathcal{P}_{\mathbf{b}|\lambda} A_w^2 \mathbf{w} \left(\frac{\mathbf{r}_i}{\lambda}, \hat{\mathbf{z}} \right) \cdot \mathbf{w} \left(\frac{\mathbf{r}_i}{\lambda}, \hat{\mathbf{z}} \right) \quad (30)$$

as it should be independent of \mathbf{b} and $\hat{\mathbf{p}}$.

The simulated time increment in a single move, $\delta \bar{t}_1 = \delta t_1 / \tau$, is then

$$\delta \bar{t}_1 = \frac{\langle \delta r_i^2 \rangle_1}{a^2} = \frac{2A_0^2 \lambda_a^2}{35N} \frac{1 - \lambda_{\min}/\lambda_{\max}}{1 - (\lambda_{\min}/\lambda_{\max})^3} \quad (31)$$

for the tapered wavelet.

F. Computational cost

We now come to calculating the computational cost for a physical time evolution of $t = \tau \bar{t}$. Here we consider a system with fractal dimension d_f , so that the expected number of particles in a move is proportional to $(\lambda/s)^{d_f}$, with s the mean separation between neighbouring particles.

The cost associated with generating move parameters is small compared to the cost of moving the n particles and calculating each of their energy changes. We can therefore use an average cost per particle per move, c , and multiply this by the expected value of n to find the total cost per move. Multiplying this by the number of moves needed to evolve by t we reach

$$C(t) \propto \frac{t}{\delta \bar{t}_1} \langle P_{\text{acc}}^{-1} \rangle c \int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \mathcal{P}_\lambda (\lambda/s)^{d_f}. \quad (32)$$

Here $\langle P_{\text{acc}}^{-1} \rangle$, defined as

$$\langle P_{\text{acc}}^{-1} \rangle \equiv \frac{\int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \mathcal{P}_\lambda \lambda^{d_f} / P_{\text{acc}}(\lambda)}{\int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda \mathcal{P}_\lambda \lambda^{d_f}}, \quad (33)$$

handles the additional λ -dependence coming from our λ recycling scheme. Its value lies between 1 and $1/P_{\text{acc}}^\infty$, itself of order unity, and limits to this upper value as $\lambda_{\max} \rightarrow \infty$. Its effect on the estimated scaling is therefore small and we will treat it as a constant.

1. Cost of homogeneous systems

When the system is homogeneous or in a poor solvent so that $d_f = 3$, the integral evaluates to a logarithm and $s^3 = L^3/N$. The cost then goes as

$$C_{\text{homo}}(t) \propto N^2 \bar{t} \frac{\langle P_{\text{acc}}^{-1} \rangle c}{A_0^2 \lambda_a^2} \left(\frac{\lambda_{\min}}{L} \right)^3 \frac{\ln(\lambda_{\max}/\lambda_{\min})}{1 - \lambda_{\min}/\lambda_{\max}}. \quad (34)$$

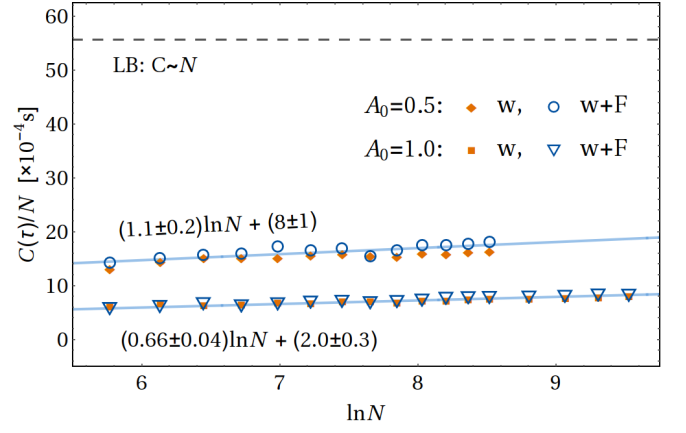


FIG. 3. CPU cost per particle to evolve semi-dilute (homogeneous) systems by a single time unit. The dashed line indicates the LB timings for Fig. 8 in Ref. 8, which used identical systems, rescaled to our time unit. Timings for both pure wavelet (w) and periodic wavelet plus Fourier (w+F) algorithms are shown for comparison. For our timings we used the gcc compiler with optimisation -O3 on a single CPU on an Intel Core2 Quad CPU Q9400 at 2.66GHz.

For systems with fixed global density and $\lambda_{\max} \sim L \sim N^{1/3}$, this reduces to $C_{\text{homo}} \sim N \ln N$.

Fig. 3 shows cost timings in a semi-dilute homogeneous system which confirms the $N \ln N$ scaling. This is true also of the wavelet plus Fourier (w+F) data in this figure, which includes moves described in section V that only add a minor cost. The following discussion therefore applies to either pure wavelet or w+F data equally.

The coefficients of the $\ln N$ term, which dictates the value of N at which it becomes cheaper to use an explicit solvent algorithm such as LB, vary with A_0 as $1/(A_0^2 P_{\text{acc}}^\infty)$. This is confirmed by reading P_{acc}^∞ from the semi-dilute data in Fig. 2, which predict a ratio of coefficients of approximately 1.6, within the error margins in Fig. 3. Note it is $(P_{\text{acc}}^\infty)^{-1}$ rather than $\langle P_{\text{acc}}^{-1} \rangle$ that is important since it is the addition of larger λ moves that contributes the $\ln N$ term, and these see P_{acc}^∞ .

A fair comparison to the LB cost shown here would use the same hardware, but the current data can give an rough estimate of how large N will be before LB is cheaper. The $A_0 = 0.5$ line crosses the dashed LB line at $\ln N \approx 43$, or $N \sim 10^{19}$. Even accounting for shifts in this value once the same hardware is used, this system size is impractically large for any simulation, especially as we foresee improvements to the wavelet code simply by using more powerful hardware and compilers, as well as the possibility of parallelising the algorithm.

2. Cost of fractal systems

When $d_f < 3$, as in the case for a single polymer chain in good ($d_f = 1/0.588$) or θ ($d_f = 2$) solvents,²⁷ the cost

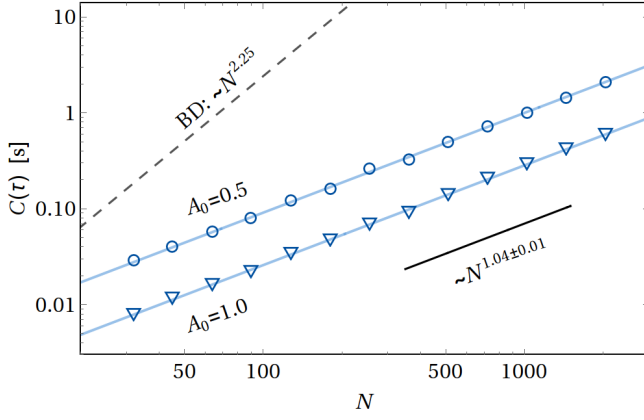


FIG. 4. CPU cost to evolve systems with an isolated polymer of length N by a single time unit with the infinite wavelet plus Fourier algorithm. The systems considered were identical to those in Fig. 11 in Ref. 6, and the dashed line indicates the BD timings from that plot, rescaled to our time unit.

integrates to

$$C_{\text{frac}}(t) \propto Nt \frac{\langle P_{\text{acc}}^{-1} \rangle c}{A_0^2 \lambda_a^2} \left(\frac{\lambda_{\min}}{s} \right)^{d_f} \frac{1 - (\lambda_{\min}/\lambda_{\max})^{3-d_f}}{1 - \lambda_{\min}/\lambda_{\max}}. \quad (35)$$

This time s is taken as the bond length between beads and is set by the potentials between particles. For θ solvents the λ_{\max} dependence cancels, while it is only significant for small systems in good solvent. In either case we are left with $C_{\text{frac}} \sim N$.

This is shown in Fig. 4. Again the differences in hardware should not be forgotten, but in the dilute regime the wavelet algorithm is seen to be much faster than conventional BD, while the LB timings were a couple of orders of magnitude higher than those for BD, with cost scaling as $N^{1.78 \approx 3/d_f}$.

G. Effect of λ_{\min} and λ_{\max} on the mobility tensor

So far the effects of using the finite bounds λ_{\min} , λ_{\max} have only been stated without proof. To see the effects we first take the Fourier transform of the wavelet representation of the Oseen tensor, as per Appendix A, substitute in the wavelet form in Eq. (20) and impose the finite limits on the λ integral. This results in the Fourier space tensor

$$\tilde{\mathcal{O}}_w(\mathbf{k}; \lambda_{\min}, \lambda_{\max}) = \frac{4\pi A_0^2}{3k^2} (\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) \int_{k\lambda_{\min}}^{k\lambda_{\max}} dq q^3 \tilde{\phi}(q)^2. \quad (36)$$

The inverse FT then gives the tensor simulated by the algorithm in a form that more clearly shows its characteristics. The inverse FT with limits $(k\lambda', \infty)$ is found in

Appendix B to be

$$\mathcal{O}'(\mathbf{r}; \lambda') = \frac{A_0^2}{3\pi r} \int_0^\infty dq q^3 \tilde{\phi}(q)^2 \left[(\mathbf{I} + \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}) \text{Si } Q + (\mathbf{I} - 3\hat{\mathbf{r}} \otimes \hat{\mathbf{r}}) \frac{\sin Q - Q \cos Q}{Q^2} \right], \quad (37)$$

where $Q = qr/\lambda'$ and Si is the sine-integral function $\text{Si } Q = \int_0^Q dt \sin t/t$. The full tensor is

$$\mathcal{O}_w(\mathbf{r}; \lambda_{\min}, \lambda_{\max}) = \mathcal{O}'(\mathbf{r}; \lambda_{\min}) - \mathcal{O}'(\mathbf{r}; \lambda_{\max}). \quad (38)$$

This tensor has approximately the Oseen form between λ_{\min} and λ_{\max} , with regularisation of the singularity below λ_{\min} and missing correlations for separations larger than λ_{\max} .

To calculate the regularisation at $r_{ij} \rightarrow 0$, and hence find the particle hydrodynamic radius, we consider the case where $\lambda_{\max} \rightarrow \infty$. This choice anticipates the modification in section V. We then associate Eq. (37) with the self- and cross-terms in section III, taking the limits $r \rightarrow 0$ and $r \rightarrow \infty$ respectively. In the former limit, the square brackets in Eq. (37) become $(4/3)Q\mathbf{I}$, while for the latter they become $(\pi/2)(\mathbf{I} + \hat{\mathbf{r}} \otimes \hat{\mathbf{r}})$. Since this yields the expected tensor structures it is sufficient to equate the scalar factors and then solve the 2 equations.

This gives the ratio

$$\frac{\lambda_{\min}}{a} = \lambda_a \equiv \frac{2 \int_0^\infty dq q^3 \tilde{\phi}(q)^2}{\pi \int_0^\infty dq q^4 \tilde{\phi}(q)^2}, \quad (39)$$

meaning λ_{\min} is exactly proportional to particle radius as previously claimed so long as the integrals exist. For the tapered wavelet, $\lambda_a = 2.316$.

Fig. 5 shows the radial element of the simulated mobility tensor, measured from correlations in particle displacements, plotted alongside curves derived from Eq. (38). The pure wavelet data shows long range deviation from r_{ij}^{-1} behaviour, originating from having a finite λ_{\max} . Intuitively this must happen as any particles separated by a distance larger than $2\lambda_{\max}$ cannot possibly have correlated motion due to never being in the same move. The w+F data shows the adaptation described shortly corrects for this as required.

Note also that the correlations are close to those of the RP tensor. As the RP tensor is already an approximation at small r_{ij} , modifying the method to explicitly replicate this tensor is not expected to be worthwhile, although we note that it is possible using Faxén's laws.³⁰

V. ADAPTATION TO INCLUDE FOURIER MOVES

We now describe how to modify the algorithm to correct for finite λ_{\max} by adding $\mathcal{O}'(\mathbf{r}; \lambda_{\max})$ back into Eq. (38). This is achieved using essentially the same approach but using plane waves instead of wavelets.

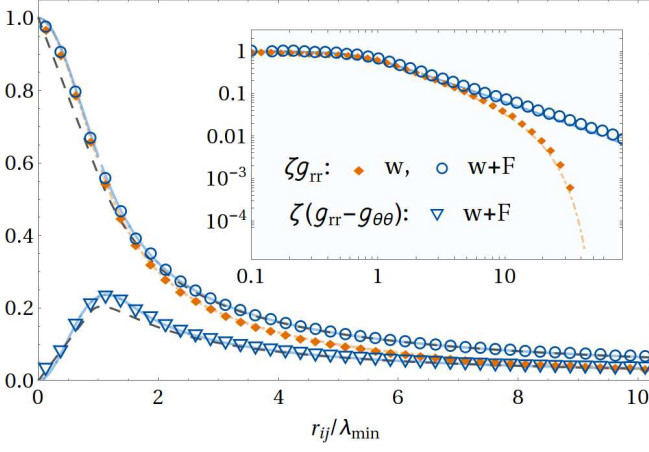


FIG. 5. Plots of simulated mobility tensor elements normalised by $\zeta = 6\pi\eta a$ in both the pure wavelet and infinite wavelet plus Fourier algorithms. The w+F data is universal in parameter choices, while the drop-off of the pure wavelet data depends on $\lambda_{\max}/\lambda_{\min}$, which has the value 29 here. Theoretical curves lie underneath the data and curves for the RP tensor are also shown (dashed) for comparison. The inset shows the same data on a log scale.

Although it didn't matter for wavelets because λ_{\max} ensured periodic images would not overlap, whether we are considering an infinite or periodic system is now important, and the corresponding tensors will be denoted $\mathcal{O}_F^\infty(\mathbf{r}; \lambda_{\max})$ and $\mathcal{O}_F^P(\mathbf{r}; \lambda_{\max})$. Similarly, other boundary condition dependent quantities will be distinguished with these superscripts, and equations that apply for either will leave the superscript off of these quantities.

We start by writing these tensors in the plane wave basis, which is none other than the Fourier transform, so we have

$$\tilde{\mathcal{O}}_F^\infty(\mathbf{k}; \lambda_{\max}) = \tilde{\mathcal{O}}'(\mathbf{k}; \lambda_{\max}), \quad (40)$$

which can be read directly from Eq. (36).

The periodic case needs to add in periodic images of this, which in position space means convolving with a Dirac comb

$$\mathcal{O}_F^P(\mathbf{r}; \lambda_{\max}) = \mathcal{O}'(\mathbf{r}; \lambda_{\max}) * \text{III}_r(\mathbf{r}); \quad (41)$$

$$\text{III}_r(\mathbf{r}) \equiv \sum_{\ell} \delta^3(\mathbf{r} - \mathbf{r}_\ell), \quad (42)$$

where $\mathbf{r}_\ell = L(\ell_x, \ell_y, \ell_z)$ with all $\ell \in \mathbb{Z}$. The convolution theorem then gives the FT

$$\tilde{\mathcal{O}}_F^P(\mathbf{k}; \lambda_{\max}) = \tilde{\mathcal{O}}'(\mathbf{k}; \lambda_{\max}) (2\pi/L)^3 \text{III}_k(\mathbf{k}) \quad (43)$$

with III_k defined similarly to III_r , but with $\mathbf{k}_\ell = (2\pi/L)(\ell_x, \ell_y, \ell_z)$ in place of \mathbf{r}_ℓ .

The inverse Fourier transforms can be re-expressed in expectation value form by recognising $(\mathbf{1} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) = 2\langle \hat{\mathbf{e}} \otimes \hat{\mathbf{e}} \rangle_{\hat{\mathbf{e}}}$, with $\hat{\mathbf{e}}$ a unit vector perpendicular to $\hat{\mathbf{k}}$. To

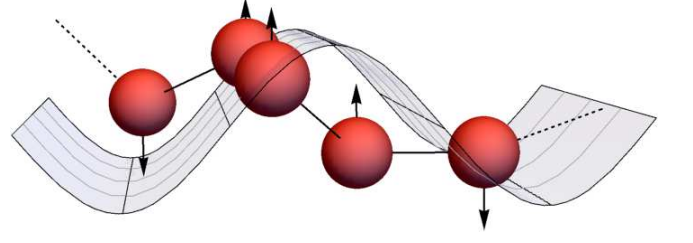


FIG. 6. Schematic diagram of how a Fourier move displaces all particles in the system. The plane wave surface indicates the vector field, which is perpendicular to \mathbf{k} and spans the entire system.

re-express $e^{i\mathbf{k} \cdot \mathbf{r}_{ij}}$ we note that $\tilde{\mathcal{O}}'(\mathbf{k}; \lambda_{\max})$ is even in \mathbf{k} , so we can replace the complex exponential with

$$\cos(\mathbf{k} \cdot \mathbf{r}_{ij}) = 2 \langle \cos(\mathbf{k} \cdot \mathbf{r}_i + \Phi) \cos(\mathbf{k} \cdot \mathbf{r}_j + \Phi) \rangle_{\Phi}, \quad (44)$$

and we finally have

$$\mathcal{O}_F(\mathbf{r}_{ij}) = \langle A_F \cos(\mathbf{k} \cdot \mathbf{r}_i + \Phi) \hat{\mathbf{e}} \otimes A_F \cos(\mathbf{k} \cdot \mathbf{r}_j + \Phi) \hat{\mathbf{e}} \rangle_{\Phi, \hat{\mathbf{e}}, \mathbf{k}}, \quad (45)$$

with A_F filling the same role as A_w did for wavelet moves. We therefore have Fourier moves causing displacements $\delta \mathbf{r} = A_F \cos(\mathbf{k} \cdot \mathbf{r}_j + \Phi) \hat{\mathbf{e}}$, as seen diagrammatically in Fig. 6. The infinite and periodic versions differ only in the hidden distributions over \mathbf{k} .

The rest of this section follows loosely the wavelet section above.

A. Choosing plane wave amplitude

Similarly to section IV C, we desire A_F to be chosen such that ΔU is independent of \mathbf{k} . Further to this, we want the change in energy to be equal to that of wavelet moves.

In a Fourier move the whole simulation box sees a displacement, so the move volume is L^3 and $n = N$. It will be seen that any L dependence cancels so this approach is still valid for the infinite case. We are free to pick $\hat{\mathbf{e}} = (1, 0, 0)$ and $\Phi = 0$ without loss of generality, so transversality forces $k_x = 0$. This simplifies the strain tensor so that the strain energy is

$$\begin{aligned} \Delta U_F &= \frac{1}{2} \mu N L^{-3} A_F^2 k^2 \int_{\text{box}} d^3 \mathbf{r} \sin^2(\mathbf{k} \cdot \mathbf{r}) \\ &= \frac{1}{4} \mu N A_F^2 k^2. \end{aligned} \quad (46)$$

Equating this to Eq. (26), we get

$$A_F = \sqrt{2/(5N)} A_0 k^{-1}. \quad (47)$$

Fig. 7 verifies that choosing this form does indeed lead to a constant acceptance probability, at least for small

k modes which will end up dominating the distribution. Because these dominate, the small k value of P_{acc} is close to the global value, which in turn can be compared to the global values in Fig. 2. Here we see that, while the global probabilities are less than for wavelets in dilute systems, in denser systems they are close as desired.

At large k the wavelength is less than the particle separation and the moves are essentially just noise with decreasing amplitude as k increases. Hence P_{acc} rises to 1, but only after a certain value of k that is the same in all data sets.

Note that Eq. (47) diverges for the $k = 0$ mode. This mode is left out in the Ewald sum in other methods to enforce zero net force.³¹ For our algorithm, the $k = 0$ mode has no influence on dynamics within the system and by viewing the system in the centre of mass frame we can set the displacements associated with this move to zero. However, because in the periodic case this mode has a non-zero probability of being chosen, we must allow these moves to occur in order to keep the correct distribution of Fourier moves, even though they induce no change in the system. Their automatic acceptance coming from $\Delta U = 0$, as well as them being quite common, accounts for the rise in P_{acc} in the lowest k bins in Fig. 7(b).

B. Distributions of wavevectors

With the amplitude chosen, we are now able to determine the distribution of \mathbf{k} . For the infinite case this picks up k^2 from $d^3\mathbf{k}$, giving

$$\mathcal{P}_k^\infty = \mathcal{N}_k^\infty k^2 \int_{k\lambda_{\text{max}}}^\infty dq q^3 \tilde{\phi}(q)^2. \quad (48)$$

For the tapered wavelet the normalisation is exactly $\mathcal{N}_k^\infty = 15\lambda_{\text{max}}^3/(2\pi)^3$.

For the periodic case, III_k turns the PDF into the discrete set of probabilities

$$P_{\mathbf{k}}^P(\mathbf{k}_\ell) = \mathcal{N}_{\mathbf{k}}^P \int_{k_\ell\lambda_{\text{max}}}^\infty dq q^3 \tilde{\phi}(q)^2, \quad (49)$$

where the normalisation, obtained by a sum of this integral over all \mathbf{k}_ℓ , cannot be related simply to λ_{max} .

For concreteness, the remaining PDFs are: $\mathcal{P}_{\hat{\mathbf{k}}}^\infty = 1/4\pi$, $\mathcal{P}_{\hat{\mathbf{e}}} = \mathcal{P}_\Phi = 1/2\pi$.

C. The probability of making a Fourier move

It now only remains to calculate how often to make a Fourier move. For this we require that the Fourier moves evolve the system by the same amount of time as the

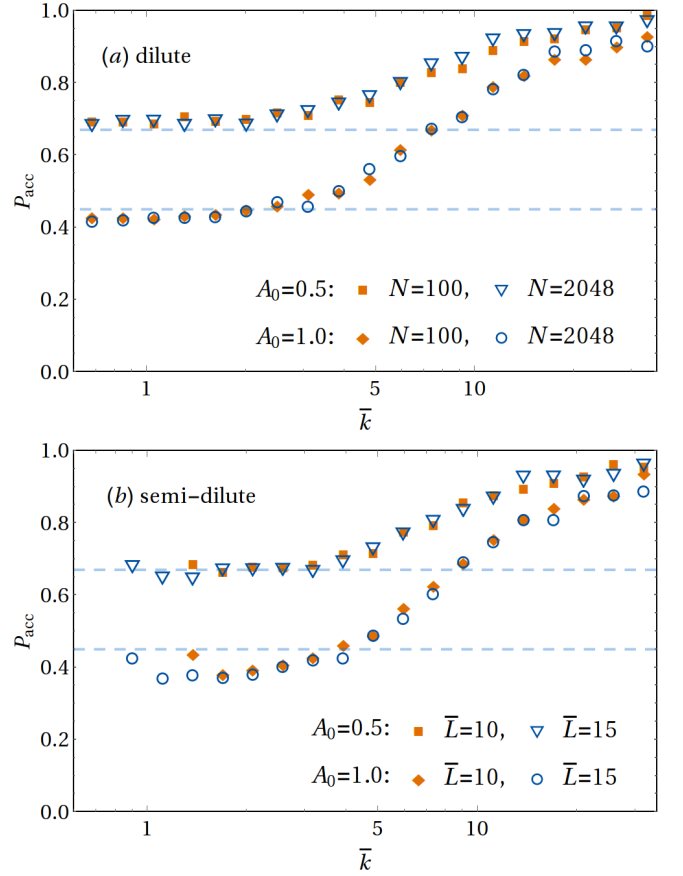


FIG. 7. Acceptance probabilities over the spectrum of plane waves different move amplitudes. (a): Single chain data using the infinite algorithm. (b): Data for semi-dilute systems with the periodic algorithm. Dashed lines indicate the global P_{acc} values for semi-dilute systems in Fig. 2.

missing wavelets with $\lambda > \lambda_{\text{max}}$ would have done. Because these wavelets lead to overlapping images in the periodic case, and hence render direct calculation of $\langle \delta r_i^2 \rangle$ infeasible, we instead observe that the tensors generated by plane waves and large wavelets are identical up to a constant. This constant is built up from the normalisation factors introduced in converting from the initially equal integral forms to expectation values, including the factors of 2 added with the introduction of both Φ and $\hat{\mathbf{e}}$. Further to these we must multiply by $(5N/2) \times (4\pi/3)$ to account for constants added and missed when A_F was fixed, and finally the infinite and periodic Fourier cases pick up $(2\pi)^{-3}$ and L^{-3} respectively from the inverse Fourier transform.

All of this together leads to the remarkably simple ratio $\mathcal{R}^\infty \equiv \langle \delta r_i^2 \rangle_F^\infty / \langle \delta r_i^2 \rangle_w^{\lambda > \lambda_{\text{max}}} = 1/2$ for the tapered wavelet. For other wavelet choices only \mathcal{N}_k^∞ changes and, as this is always proportional to λ_{max}^3 , the ratio is always a pure number.

The ratio for the periodic case does not need to change anything from the wavelet part on the assumption that the convolution is included in the tensor in its entirety,

which is consistent with the previous analysis as the convolution does not change small λ wavelets. The ratio is therefore

$$\mathcal{R}^P = \left(\frac{L}{2\pi}\right)^3 \frac{\mathcal{N}_{\mathbf{k}}^P}{\mathcal{N}_k^\infty \mathcal{N}_{\mathbf{k}}^\infty} \mathcal{R}^\infty = \frac{2\pi}{15} \left(\frac{L}{\lambda_{\max}}\right)^3 \mathcal{N}_{\mathbf{k}}^P. \quad (50)$$

To calculate the probability of making a Fourier move, P_F , imagine that in the pure wavelet case, with $\lambda_{\max} \rightarrow \infty$, we make X moves, and in the w+F case we make $X(1 + \Delta)$ moves for the same time evolution. We do not want the number of wavelet moves with $\lambda < \lambda_{\max}$ to change, so we must have

$$(1 - P_F) X(1 + \Delta) = (1 - (\lambda_{\min}/\lambda_{\max})^3) X, \quad (51)$$

while enforcing equal time evolution for large wavelets and plane waves requires

$$\mathcal{R} P_F X(1 + \Delta) = (\lambda_{\min}/\lambda_{\max})^3 X. \quad (52)$$

Here $(\lambda_{\min}/\lambda_{\max})^3$ is the probability of picking a wavelet with $\lambda > \lambda_{\max}$.

Clearly X cancels so that only Δ and P_F are unknown. Solving for the latter and simplifying gives us

$$P_F = (1 + \mathcal{R} [(\lambda_{\max}/\lambda_{\min})^3 - 1])^{-1}. \quad (53)$$

For typical systems and wavelet bounds this is several orders of magnitude less than 1.

Finally we use this to calculate the new value of δt_1 , which requires a direct calculation of $\langle \delta r_i^2 \rangle_F^\infty$:

$$\langle \delta r_i^2 \rangle_F^\infty = \frac{A_0^2 \mathcal{N}_k^\infty}{5N} \int_0^\infty dk \int_{k\lambda_{\max}}^\infty dq q^3 \tilde{\phi}(q)^2 = \frac{A_0^2 \lambda_{\max}^2}{35N}. \quad (54)$$

The time increment is then simply the appropriate weightings of wavelet and Fourier self-diffusion divided by a^2 ,

$$\frac{\delta t_1}{\tau} = \frac{A_0^2 \lambda_a^2}{35N \lambda_{\min}^2} \left(2 \frac{\lambda_{\min}^{-1} - \lambda_{\max}^{-1}}{\lambda_{\min}^{-3} - \lambda_{\max}^{-3}} (1 - P_F^\infty) + \lambda_{\max}^2 P_F^\infty \right), \quad (55)$$

for *both* the infinite and periodic cases since τ is defined for an isolated particle in an infinite system.

D. Computational cost

Armed with the frequency at which Fourier moves are chosen, we now calculate the computational cost associated with them. Each plane wave evolves all N particles, each with a cost c as in the wavelet case, by $\delta t_1 \sim N^{-1}$. For Fourier moves alone then, the cost scales as N^2 .

Accounting for the presence of wavelet moves, the Fourier contribution to the cost is weighted by $P_F \sim 1/N$ for a set of constant density systems. The Fourier moves are therefore subdominant to the wavelet cost, and the previous $N \ln N$ scaling remains, as observed in Fig. 3. Similarly the fractal scaling of N is unchanged.

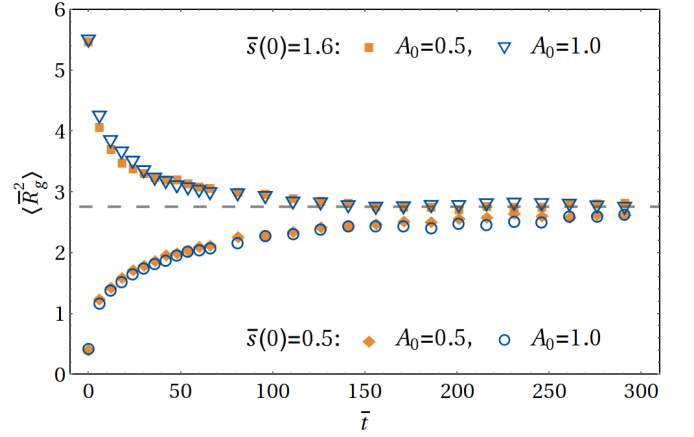


FIG. 8. Relaxation of an isolated polymer with $N_b = 10$ and WCA and FENE potentials, for different A_0 and initial bond lengths, $\bar{s}(0)$. The dashed line shows the value for the same system in Ref. 8.

VI. ALGORITHM VALIDATION

A. Chain relaxation

Relaxation of isolated chains of length $N_b = 10$ beads with initially stretched or compressed bonds is seen in Fig. 8, with mean-square radius of gyration defined as

$$\langle R_g^2 \rangle = \frac{1}{2N_b^2} \sum_{i=1}^{N_b} \sum_{j=1}^{N_b} \langle r_{ij}^2 \rangle. \quad (56)$$

Both stretched and compressed chains relax to the same value as obtained for this system in Ref. 8, confirming the algorithm equilibrates correctly. The figure also shows the same physical relaxation time for the different values of A_0 , verifying the A_0 dependence in Eq. (31).

B. Chain diffusivity

Diffusion of the centre of mass of chains of different lengths is shown in Fig. 9(a), where the correct linear relationship with time is seen. Note that we are ignoring the difference between short and long time diffusion in the analysis as this basic data is not precise enough to distinguish the few percent between them.³² Data at that accuracy is left for future work.

Fig. 9(b) shows measured centre of mass diffusion constants, $D = \langle \Delta r_{\text{CoM}}^2 \rangle / 6t$, for chains in both good and θ solvents, with the former using the same data as in Fig. 9(a). For the θ solvent the WCA potential was turned off and the FENE potential was replaced with the harmonic bond potential $(3/2)k_{\text{FENE}}(r_{ij} - r_0)^2$, in which $\bar{r}_0 = 0.9691$ was used to match the mean bond length in the good solvent. These sets of data show quantitatively that simulations in both types of solvent lead to

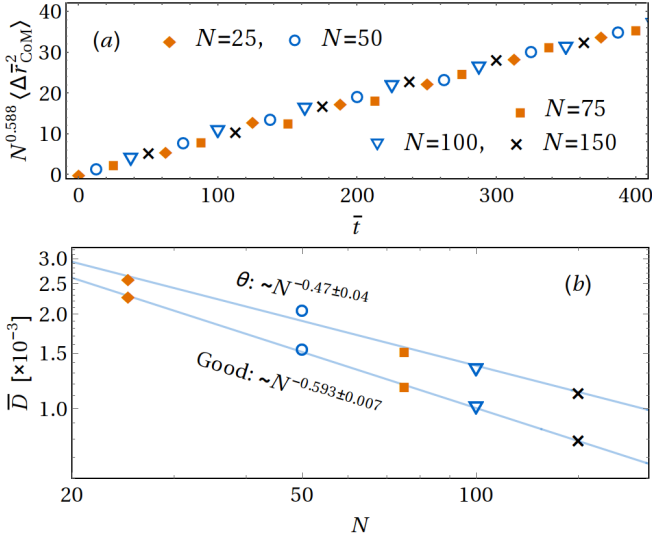


FIG. 9. (a): Mean centre of mass displacement squared for isolated chains in a good solvent, scaled to show the theoretical collapse of the data. Every fifth data point has been used from each data set to show this clearly. (b): Measured diffusion constants plotted against chain length for chains with identical mean bond lengths in good and θ solvents.

expected scaling with N within the error bounds on the exponent.²⁷

This scaling is usually considered asymptotic, requiring long chains to observe. This is because of correction terms in the diffusivity³³

$$D = \frac{k_B T}{6\pi\eta a} [N^{-1} + a (AN^{-\nu} - BN^{-1} + \dots)], \quad (57)$$

where the first term, coming from the sum of hydrodynamic radii of individual monomers, is often neglected. To estimate how large we would expect the corrections to be, in θ solvent with $r_0 = 0$ (i.e. Gaussian chains) it was found that $B = 4.036/s$,³³ with s the root mean squared bond length. For our systems $s \approx 1$ and $a \approx 0.3$, so $aB \approx 1.2$ and there is significant cancellation with the first term. Meanwhile $aA \approx 1.1$ so the leading correction term is actually quite small, leading to the observed scaling at small N .

For polymers in good solvent there are additional ‘non-analytic’ terms that come from excluded volume interactions. Nonetheless the qualitative result is the same when we use coefficients fitted in Ref. 33, and the observed scaling in our systems is expected to be accurate after $N \sim 50$. We do not expect the coefficients used here to apply exactly for our systems, but they do explain the scaling observed.

C. Finite box effect on self-diffusion

The periodic algorithm should lower the diffusivity because of the extra HIs with periodic images. The simplest

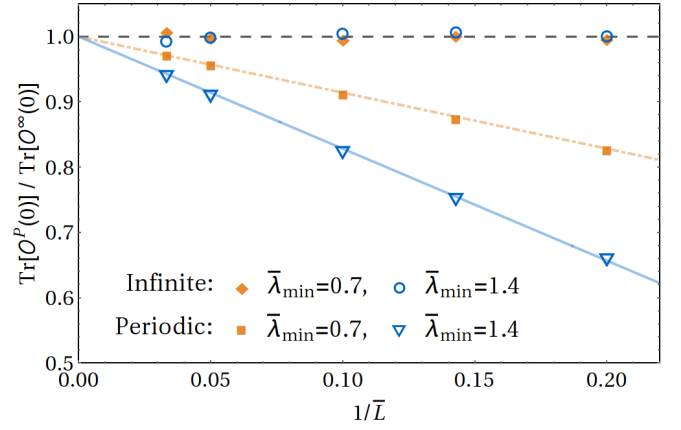


FIG. 10. Simulated traces of self-diffusion tensors, scaled by the mean trace of infinite systems for each λ_{\min} , in simulation boxes with \bar{L} between 5 and 30. The solid and dot-dashed lines were calculated using Eq. (58).

check for this is to look at the trace of the mobility tensor in the limit $r_{ij} \rightarrow 0$, which has been calculated to leading order to vary with box size as³⁴

$$\text{Tr}[\mathcal{O}^P(0; \lambda_{\min})] = \text{Tr}[\mathcal{O}^\infty(0; \lambda_{\min})] \left(1 - 2.837 \frac{a}{\bar{L}}\right). \quad (58)$$

By generating correlation curves for $\langle \delta r_i^2 \rangle$ with non-interacting particles, similar to Fig. 5, and comparing the unscaled values at $r_{ij} \rightarrow 0$, we have confirmed this behaviour as well as the L -independence of the infinite algorithm (see Fig. 10). This figure also confirms that Eq. (55) is indeed appropriate for periodic systems as the measured diffusivities are inversely proportional to how we scale time. Finally, it provides another verification of Eq. (39) as this determines the gradient of the lines.

VII. SUMMARY AND OUTLOOK

In this paper we have detailed a Monte Carlo algorithm that uses wavelet and plane wave moves to produce hydrodynamics without explicit decomposition of the mobility tensor. Our algorithm has been shown to approximate the Rotne-Prager tensor by starting from the Oseen tensor and removing small radii moves, which in turn provides a particle radius. The distribution of additional Fourier moves, meanwhile, can be chosen to simulate either a periodic or infinite system with the main wavelet part of the algorithm left unchanged.

It has been shown that the computational cost of this algorithm scales well to large systems, going as $N \ln N$ for homogeneous systems with fixed particle density and N for fractal systems. Both of these have a very competitive prefactor, making it a very promising simulation technique for extending the reach of soft-matter simulations.

The algorithm has been tested to show correct hydrodynamic correlations are produced, while the simulated

behaviour of isolated polymer chains has been shown to agree with previous work.

With the core set out, we plan to develop the algorithm further to include solvent flows and a variety of particle interactions, as well as running the code in parallel by making multiple, non-overlapping wavelet moves simultaneously. There is also scope for optimisation of what already exists. For instance, the effect of the choice of mother wavelet, and correspondingly the value of λ_a , is yet to be explored. Finally, we envision possible efficiency gains by using smart MC, in which moves are biased according to the force conjugate to them.

ACKNOWLEDGMENTS

This work has been supported by the Engineering and Physical Sciences Research Council (EPSRC), grant number EP/L505110/1 (OTD), and the Monash-Warwick Alliance (RCB). The authors acknowledge stimulating discussions with J R Prakash and are also grateful to the remaining organisers of “Hydrodynamic Fluctuations in Soft-Matter Simulations” CECAM/Monash workshop, Prato (February 2016) wherein many helpful discussions were had.

Appendix A: Showing the wavelet representation gives the Oseen tensor

Here we evaluate the integral in Eq. (17) to show that this reduces to a form exactly proportional to the Oseen tensor. We begin by taking the Fourier transform with respect to $\mathbf{r}_i - \mathbf{r}_j$, leading to

$$\begin{aligned}\tilde{g}(\mathbf{k}) &= \int d^3\mathbf{r} e^{-i\mathbf{k}\cdot(\mathbf{r}_i - \mathbf{r}_j)} g(\mathbf{r}_i - \mathbf{r}_j) \\ &= \mathcal{N}_g \int d^2\hat{\mathbf{p}} \frac{d\lambda}{\lambda} \lambda^2 \tilde{\mathbf{w}}(\lambda\mathbf{k}, \hat{\mathbf{p}}) \otimes \tilde{\mathbf{w}}(-\lambda\mathbf{k}, \hat{\mathbf{p}}),\end{aligned}\quad (\text{A1})$$

where $\tilde{\mathbf{w}}(\mathbf{k}, \hat{\mathbf{p}}) = \int d^3\mathbf{x} e^{-i\mathbf{k}\cdot\mathbf{x}} \mathbf{w}(\mathbf{x}, \hat{\mathbf{p}})$ is the Fourier transform of the mother wavelet at fixed $\hat{\mathbf{p}}$.

We next integrate over the wavelet polarisation to obtain

$$\tilde{g}(\mathbf{k}) = 4\pi\mathcal{N}_g \int d\lambda \lambda \mathbf{W}(\lambda\mathbf{k}); \quad (\text{A2})$$

$$\begin{aligned}\mathbf{W}(\mathbf{k}) &= \frac{1}{4\pi} \int d^2\hat{\mathbf{p}} \tilde{\mathbf{w}}(\mathbf{k}, \hat{\mathbf{p}}) \otimes \tilde{\mathbf{w}}(-\mathbf{k}, \hat{\mathbf{p}}) \\ &= (\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) f(k),\end{aligned}\quad (\text{A3})$$

which has the explicit tensor structure shown because the constraint $\nabla \cdot \mathbf{w} = 0$ leads to $\mathbf{k} \cdot \mathbf{W} = 0$. The amplitude factor is set by

$$2f(k) = \text{Tr}[\mathbf{W}(\mathbf{k})] = \frac{1}{4\pi} \int d^2\hat{\mathbf{p}} \tilde{\mathbf{w}}(\mathbf{k}, \hat{\mathbf{p}}) \cdot \tilde{\mathbf{w}}(-\mathbf{k}, \hat{\mathbf{p}}). \quad (\text{A4})$$

Reassembling all this and making the substitution $q = \lambda k$ leads to

$$\tilde{g}(\mathbf{k}) = 4\pi\mathcal{N}_g (\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) k^{-2} \int dq q f(q). \quad (\text{A5})$$

This exactly matches the Fourier transform of the Oseen tensor, $(1/\eta k^2)(\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}})$, upon choosing the normalisation $\mathcal{N}_g^{-1} = 4\pi\eta \int dq q f(q)$.

Appendix B: Derivation of $\mathcal{O}'(r; \lambda')$

The inverse Fourier transform of Eq. (36), with the q integral from $k\lambda'$ to ∞ , is:

$$\begin{aligned}\mathcal{O}'(r; \lambda') &= \\ \frac{A_0^2}{6\pi^2} \int_{\text{all } \mathbf{k}} d^3\mathbf{k} e^{i\mathbf{k}\cdot\mathbf{r}} k^{-2} (\mathbf{I} - \hat{\mathbf{k}} \otimes \hat{\mathbf{k}}) \int_{k\lambda'}^{\infty} dq q^3 \tilde{\phi}(q)^2.\end{aligned}\quad (\text{B1})$$

First, we commute the q and k integrals, which requires a change in the limits. To be able to perform the integral we also note that $\hat{\mathbf{k}} \otimes \hat{\mathbf{k}} e^{i\mathbf{k}\cdot\mathbf{r}} = -\nabla \otimes \nabla k^{-2} e^{i\mathbf{k}\cdot\mathbf{r}}$. What remains is an inverse FT of a radial function, so the angular integrals over the exponential give the usual result of $4\pi \sin(kr)/kr$, so that we have

$$\begin{aligned}\mathcal{O}'(r; \lambda') &= \\ \frac{2A_0^2}{3\pi} \int_0^{\infty} dq q^3 \tilde{\phi}(q)^2 \int_0^{q/\lambda'} dk (\mathbf{I} + \nabla \otimes \nabla k^{-2}) \frac{\sin kr}{kr}.\end{aligned}\quad (\text{B2})$$

We now focus on performing the k integral.

The \mathbf{I} term integrates directly to $\text{Si}(qr/\lambda') \mathbf{I}/r$. The other term appears to diverge at the $k \rightarrow 0$ limit, but it can be split up into an integral with limits $(0, \infty)$ minus a finite one over $(q/\lambda', \infty)$. Although the first of these still appears to diverge, it is exactly what would have been integrated if we had the full Oseen tensor. It is therefore known to be $(\pi/4r)(\hat{\mathbf{r}} \otimes \hat{\mathbf{r}} - \mathbf{I})$.²⁷

Hence we are only left with $-\nabla \otimes \nabla$ acting on

$$\begin{aligned}\frac{1}{r} \int_{q/\lambda'}^{\infty} dk k^{-3} \sin(kr) &= \\ \frac{\lambda'}{2q} \left[\cos Q + Q^{-1} \sin Q + Q \left(\text{Si} Q - \frac{\pi}{2} \right) \right],\end{aligned}\quad (\text{B3})$$

with $Q = qr/\lambda'$. Performing the two derivatives obtains the form

$$\begin{aligned}\left(\text{Si} Q - \frac{\pi}{2} \right) \frac{1}{2r} (\hat{\mathbf{r}} \otimes \hat{\mathbf{r}} - \mathbf{I}) \\ + \frac{\sin Q - Q \cos Q}{Q^2} \frac{1}{2r} (\mathbf{I} - 3\hat{\mathbf{r}} \otimes \hat{\mathbf{r}}).\end{aligned}\quad (\text{B4})$$

The $\pi/2$ term cancels exactly with the full Oseen term above, while the remaining terms plus the contribution from the first \mathbf{I} term give the final result

$$\begin{aligned} \mathcal{O}'(\mathbf{r}; \lambda') = & \frac{A_0^2}{3\pi r} \int_0^\infty dq q^3 \tilde{\phi}(q)^2 \left[(\mathbf{I} + \hat{\mathbf{r}} \otimes \hat{\mathbf{r}}) \text{Si } Q \right. \\ & \left. + (\mathbf{I} - 3\hat{\mathbf{r}} \otimes \hat{\mathbf{r}}) \frac{\sin Q - Q \cos Q}{Q^2} \right]. \quad (\text{B5}) \end{aligned}$$

Note that this calculation could have started from Eq. (A5) for a more general wavelet. In this case, all that needs changing are $q^2 \tilde{\phi}(q)^2 \rightarrow f(q)$ and the constants at the front.

- ¹D. L. Ermak and J. A. McCammon, The Journal of Chemical Physics **69**, 1352 (1978).
- ²K. Kremer and G. S. Grest, The Journal of Chemical Physics **92**, 5057 (1990).
- ³P. J. Hoogerbrugge and J. M. V. A. Koelman, EPL (Europhysics Letters) **19**, 155 (1992).
- ⁴A. J. C. Ladd, Journal of Fluid Mechanics **271**, 285 (1994).
- ⁵A. J. C. Ladd, Journal of Fluid Mechanics **271**, 311 (1994).
- ⁶T. T. Pham, U. D. Schiller, J. R. Prakash, and B. Dünweg, The Journal of Chemical Physics **131**, 164114 (2009).
- ⁷A. Malevanets and R. Kapral, The Journal of Chemical Physics **110**, 8605 (1999).
- ⁸A. Jain, S. P., B. Dünweg, and J. R. Prakash, Physical Review E **85**, 066703 (2012).
- ⁹M. Fixman, Macromolecules **19**, 1195 (1986).
- ¹⁰A. Saadat and B. Khomami, The Journal of Chemical Physics **140**, 184903 (2014).
- ¹¹J. P. Hernández-Ortiz, J. J. de Pablo, and M. D. Graham, The Journal of Chemical Physics **125**, 164906 (2006).
- ¹²A. Sierou and J. F. Brady, Journal of Fluid Mechanics **448**, 115 (2001).
- ¹³A. J. Banchio and J. F. Brady, The Journal of Chemical Physics **118**, 10323 (2003).
- ¹⁴A. Saadat and B. Khomami, Phys. Rev. E **92**, 033307 (2015).
- ¹⁵S. Nedelcu and J.-U. Sommer, The Journal of Chemical Physics **130**, 204902 (2009).
- ¹⁶L. Dodd, T. Boone, and D. Theodorou, Molecular Physics **78**, 961 (1993).
- ¹⁷M. R. Betancourt, The Journal of Chemical Physics **123**, 174905 (2005).
- ¹⁸S. Zamuner, A. Rodriguez, F. Seno, and A. Trovato, PLoS ONE **10**, 1 (2015).
- ¹⁹A. Uhlherr, V. G. Mavrantzas, M. Doxastakis, and D. N. Theodorou, Macromolecules **34**, 8554 (2001).
- ²⁰N. C. Karayiannis, A. E. Giannousaki, and V. G. Mavrantzas, The Journal of Chemical Physics **118**, 2451 (2003).
- ²¹E. P. Bernard, W. Krauth, and D. B. Wilson, Phys. Rev. E **80**, 056704 (2009).
- ²²T. A. Kampmann, H.-H. Boltz, and J. Kierfeld, The Journal of Chemical Physics **143**, 044105 (2015).
- ²³A. C. Maggs, Phys. Rev. Lett. **97**, 197802 (2006).
- ²⁴A. Grossmann and J. Morlet, SIAM Journal on Mathematical Analysis **15**, 723 (1984).
- ²⁵J. van den Berg, ed., *Wavelets in Physics* (Cambridge University Press, 2004).
- ²⁶C. Heil and D. F. Walnut, eds., *Fundamental Papers in Wavelet Theory* (Princeton University Press, 2006).
- ²⁷M. Doi and S. Edwards, *The Theory of Polymer Dynamics*, International series of monographs on physics (Clarendon Press, 1986).
- ²⁸R. Zwanzig, J. Kiefer, and G. H. Weiss, Proceedings of the National Academy of Sciences in the United States of America **60**, 381 (1968).
- ²⁹J. Rotne and S. Prager, The Journal of Chemical Physics **50**, 4831 (1969).
- ³⁰L. Durlinsky, J. F. Brady, and G. Bossis, Journal of Fluid Mechanics **180**, 21 (1987).
- ³¹C. W. J. Beenakker, The Journal of Chemical Physics **85**, 1581 (1986).
- ³²B. Liu and B. Dünweg, The Journal of Chemical Physics **118**, 8061 (2003).
- ³³B. Dünweg, D. Reith, M. Steinhauser, and K. Kremer, The Journal of Chemical Physics **117**, 914 (2002).
- ³⁴B. Dünweg and K. Kremer, The Journal of Chemical Physics **99**, 6983 (1993).